An Easy Protocol—How to Glitch Image Generating AI Systems

by Michael Betancourt

The first thing to keep in mind when glitching an AImage generator is that there is *no* glitch if the machine generates a version of the expected image, even if it looks "glitchy." Most traditional glitches in Glitch Art are products of compression errors and so the artifacts of compression—macroblocks, smears, etc.—are a result of the compression malfunctioning and not actually producing the image that was encoded. AI systems are generative, so glitches in these systems look qualitatively different. The unexpected and distorted outputs AI creates are not problems, but ambiguities in the data handling which create what look like problems to humans, but there is no glitch in the machine or its data. To glitch AI, we need to create a situation where the system itself breaks and does not generate an output properly, as when the compression malfunctions. *This is the protocol I use in my own work with glitching AImages.*

This protocol can be used with any AI generator that runs locally, on a personal machine, but does not work with website-based systems such as Midjourney because to glitch AI requires control over what data is being used.

The protocol:

1. **You will need to run a local AI generative system.** There are many apps available. Pick one you like. What matters is control over the data, not the system running it.

2. **You will need to collect a variety of LoRA Models** from places such as Civit.AI that allow user-generated models. Look for highly specialized models that only do one thing. These tend to exhibit what is called "overfit" and will produce errors when used together with other LoRAs. You will need between 5 and 6 of them at a minimum. (More experienced users can use their own data, but for this protocol it does not matter so long as the LoRAs produce "overfit".)

3. **Load all of these LoRAs into your AI generative system.** The main training checkpoint (safetensor) data is less important than the LoRAs for this protocol. Conflicts between the LoRAs produce the image.

4. **Apply all the "overfit" LoRAs to an image generation prompt and run the process.** The prompt is actually irrelevant; however, some LoRAs require specific activation prompts so you will need to include those to activate those LoRAs. The prompt used will not otherwise matter, since the image is produced by the conflicts between the LoRAs.

5. **You can control the resulting output by changing the weight of individual LoRAs, or changing the diffusion seed value.** These are system level controls that adjust the image apart from the prompt. Try several versions of your image: when these controls are the only thing that changes the output you have successfuly glitched the AI system!

Beware: do not be surprised if the gives you offensive or uncomfortable NSFW images as outputs. These are produced by most AI systems, but are filtered by the commercial systems so you never see them.